

Find the Cat

Projet RS

Rédigé par  
*Trabelsi Youssef*  
*Marsou Soulaïman*

Rapport

16 décembre 2022

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Choix de conception</b>	<b>3</b>
<b>3</b>	<b>Organisation</b>	<b>3</b>
<b>4</b>	<b>Fonctionnalités principales</b>	<b>3</b>
<b>5</b>	<b>Extensions réalisées</b>	<b>4</b>
5.1	Recherche par date à l'aide de mot clé . . . . .	4
5.2	Option -perm . . . . .	4
5.3	Option -ou . . . . .	4
<b>6</b>	<b>Tests</b>	<b>5</b>
<b>7</b>	<b>Difficultés rencontrées</b>	<b>5</b>
<b>8</b>	<b>Nombre d'heures</b>	<b>5</b>
<b>9</b>	<b>Conclusion</b>	<b>5</b>

# 1 Introduction

Ce projet a été réalisé dans le cadre du module RS .

L'objectif de ce projet est de réaliser un programme qui permet de retrouver un fichier dans une arborescence avec des options précises.

Le travail s'est donc découpé en plusieurs parties :

- Conception
- Implémentation
- Tests
- Rédaction du rapport

## 2 Choix de conception

La phase de conception a permis de faciliter le travail en ayant une bonne compréhension du livrable attendu. Puisque l'objectif du programme est de parcourir les fichiers puis de les filtrer pour en afficher certains, nous avons décidé d'écrire une fonction qui parcourt tous les fichiers (fonction `find()`). Durant ce parcours, une fonction de filtre analyse les options pour choisir si oui ou non tel fichier doit être affiché ou non. Cette fonction de filtrage fait appel à plusieurs fonction de filtre (`filterName`, `filterSize`, `filterMime`, etc... )pour analyser le fichier sur lequel il faut prendre une décision en fonction de chaque option. Pour la propreté du code, nous avons séparé chaque fonction dans différents fichiers.

## 3 Organisation

Lors de l'affectation des groupes, notre groupe a décidé de diviser les tâches. Chaque personne devait faire certaines questions et on se concertait si il y avait besoin d'aide.

Notre objectif, au début, était de finir au moins la partie "travail à réaliser".

Un de nos objectif pour atteindre ce but était d'essayer de valider les tests automatiques. Un objectif qui semble être atteint d'après la pipeline du git.

## 4 Fonctionnalités principales

Dans notre projet, nous avons développé toutes les fonctionnalités de base.

### 1. Parser les options et option test - Youssef

Comme indiqué dans le sujet cette option va renvoyer une certaine valeur de flag.

Exemple : `./ftc . -test -name txt` renvoie : La valeur du flag `-name` est `txt`.

Cette option s'exécute séparément pour ne pas nuire aux autres options.

### 2. Parcourir l'arborescence - Soulaïman

Option fonctionnelle. Nous avons ajouté un peu d'indentation pour rendre l'affichage plus lisible.

### 3. Option size - Youssef

Cette fonctionnalité permet d'afficher les fichiers dont leur tailles respectent la taille donnée comme paramètre de l'option.

Pour faire cela elle étudie la taille donnée et filtre les fichiers. Si ce fichier passe le filtre(i.e sa

taille respecte la taille demandé) il est affiché.

Pour tester cette fonction on l'a comparé avec la fonction `find -size`.

#### 4. **'ET' - Youssef**

Cette fonctionnalité permet de filtrer les fichiers à partir des options données en ligne de commande et les afficher.

Pour faire cela, on stocke les options et leurs paramètres correspondant dans un tableau adapté. Ensuite, le filtre prend en compte toutes les options passées dans ce tableau.

Exemple : `./ftc . -size +10k ET -date +3h` renvoie les fichiers dont la taille est plus grande de 10k et modifiés depuis au moins 3 heures.

#### 5. **Option date - Youssef**

Cette fonctionnalité permet d'afficher les fichiers dont leur date de dernière modification respecte le paramètre donné.

#### 6. **Option regex - Soulaïman**

Option fonctionnelle.

#### 7. **Option mime - Soulaïman**

Pour cette option, nous avons utilisé la librairie externe "Megamimes" (<https://github.com/kobbyowen/MegaMimes>).

#### 8. **Option ctc - Soulaïman**

Option fonctionnelle.

#### 9. **Option dir - Soulaïman**

Option fonctionnelle.

#### 10. **Gestion des messages d'erreur - Youssef / Soulaïman**

Pour chaque cas où il y avait erreur, nous affichons dans la sortie des messages d'erreurs (fd 2).

#### 11. **Propreté du code - Youssef / Soulaïman**

La conception initiale détaillée dans le paragraphe "Choix de conception" permet naturellement d'avoir un code propre et organisé.

## 5 Extensions réalisées

### 5.1 Recherche par date à l'aide de mot clé

Cette particularité permet d'accepter aussi les paramètres "now", "today", "yesterday" et "this month" pour l'option date.

(Remarque pour le "now", on a fait en sorte qu'on prend que les fichiers modifiés les 5 dernières minutes)

### 5.2 Option -perm

Cette option permet d'afficher les fichiers selon leur type de permission.

Les permissions recherchées sont passées avec le format octal.

Option fonctionnelle.

### 5.3 Option -ou

Cette option est complémentaire de la fonctionnalité ET.

Elle permet de faire un ou logique sur les différents options données.

Elle marche de la même manière que la fonctionnalité ET.

Option fonctionnelle.

## 6 Tests

Les tests utilisés sont les tests automatiques réalisés par les encadrants. Voici les résultats obtenu le 16 décembre :

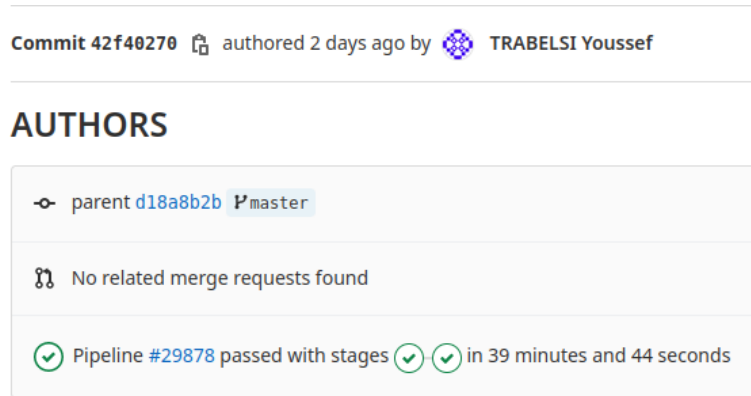


FIGURE 1 – Capture d'écran gitlab ; Résultat de la pipeline le 16 décembre

Tous les tests sont passés avec succès.

## 7 Difficultés rencontrées

Une des plus grandes difficultés rencontrées a été la contrainte de temps. Même si on a eu accès au sujet depuis fin octobre, on devait aussi travailler sur les autres projets. Pour essayer de résoudre ce problème, notre groupe a essayé de faire des réunions d'avancement durant toute la période du projet. Ces réunions nous permettaient de vérifier l'avancement du projet et les problèmes techniques rencontrés.

Un autre élément qui a été perturbant a été la modification fréquente du comportement des tests. Lorsque les fichiers tests étaient modifiés, il arrivait à notre programme de ne plus fonctionner correctement.

On a aussi eu quelques problèmes techniques. Ces problèmes étaient généralement liés à notre manque de connaissances en langage C.

## 8 Nombre d'heures

Pour la conception, nous avons discuté pendant 2h pour mettre au clair la façon dont on a voulu procéder. Par la suite, chacun a eu une part de conception à faire pour chaque fonctionnalité du programme. Les autres différentes phases de ce projet n'étaient pas dissociées. Au total nous avons passé chacun sur le projet environ :

TRABELSI Youssef : 35h

MARSOU Soulaïman : 25h

## 9 Conclusion

Le sujet a été très instructif pour l'exercice du langage C. Notre groupe a essayé de son mieux pour atteindre son objectif initial. Cependant faire tout le projet n'était pas si évident. Même si les difficultés rencontrées ralentissaient la réalisation du projet, nous avons réussi à faire toute la partie "travail à réaliser" et quelques extensions optionnelles.